```java
public void getMaxProfit() {
        int[][] table = new int[n + 1][capacity + 1];
        // fill up table
        for (int i = 1; i < n + 1; i++) {
            for (int j = 1; j < capacity + 1; j++) {
                if (weights[i - 1] > j) {
                    // item doesn't fit, so we can't pick it up
                    table[i][j] = table[i - 1][j];
                } else {
                    // max of value item picked up or not
                    table[i][j] = Math.max(table[i - 1][j], values[i - 1] + table[i - 1][j - weights[i - 1]]);
                }
            }
        }

        int[] pickedUpItems = new int[n];
        int maxValue = table[n][capacity], tempValue = maxValue, i = n, j = capacity, maxCapacity = 0;
        // traceback, discovering items picked up and computing total weight
        while (tempValue != 0) {
            if (tempValue == table[i - 1][j]) {
                // not picked up, so go one row up
                tempValue = table[i - 1][j];
                i--;
            } else {
                // item picked up, so go up and to the left as remaining capacity is decreased
                pickedUpItems[i - 1] = 1;
                maxCapacity += weights[i - 1];
                tempValue = table[i - 1][j - weights[i - 1]];
                j -= weights[i - 1];
                i--;
            }
        }
    }
```